The test and validation accuracies (final = ensemble accuracy):

```
IRT Validation Accuracy: 0.5643522438611346
NN Validation Accuracy: 0.6185436071126165
KNN Validation Accuracy: 0.6109229466553768
IRT Test Accuracy: 0.5797346881174146
NN Test Accuracy: 0.6181202370872142
KNN Test Accuracy: 0.615015523567598
Final validation accuracy: 0.6045723962743438, final test accuracy: 0.6096528365791702
```

Ensemble Process:

I first started with bootstrapping the training data. I did not sample the .csv data files since it would not make sense to sample that with replacement as duplicates wouldn't do anything. Instead, I sampled the sparse matrix's rows, which means I sampled the student's data with replacement. I chose the rows/students instead of the columns/questions since it seems all the models are trying to predict a student's correctness/are user based not item based (I did not work on the IRT model, but it seems that the model, despite the name, can be interpreted as user based or at least can be both user and item based). I chose to sample enough that the bootstrapped data have the same length as the original training set since that seems the most intuitive and simple instead of picking an arbitrary number which I have no statistics on. I created 3 bootstrapped datasets from the training data and then trained each model with a separate bootstrapped dataset.

To train, I just copied the main() function of each model into its own function to train that model and then modified it to work in our context. I also made sure to keep the hyperparameters testing (for example in kNN try out different k values and then use validation data to choose best k) so that the validation data is properly used. After training the model and choosing the best hyperparameter with the validation set, I then used the model to make predictions on both the validation and test set. I then summed up the raw predictions and divided by 3 and then apply the 0.5 threshold to get the ensemble's prediction. I can just add up the model's predictions since the training/validation data used for each is the same and are aligned, so each data point/index is on the same prediction.

Results:

The ensemble performed worse than any of the 3 base models (when those base models are run against the training set) which is surprising. Doing more debugging, I confirmed that my logic/code for averaging the predictions for the ensemble prediction is correct. I also outputted the validation and test accuracies of each of the models themselves and found out that each of the base models has lower performance when ran against the

bootstrapped data which is the cause for the decrease in performance. Double checking my code for bootstrapping the training set it seems correct, so I assume this is an issue with bootstrapping itself. I think this is because when we bootstrap/sample with replacement then we lose some diversity in the dataset as we get duplicates of some data, and we also lose some data completely as they are not chosen to be sampled. The bootstrapped data is also noisier than the original data due to the randomness of sampling. These reasons result in lower accuracy on the validation and test data. The ensemble is averaging the predictions of its base models so it is natural for the performance to decrease if the base models perform worse.